Curso Avançado de Controladores Programáveis

Ref. 5-0013.130

Manual Rev. 1.30 Março/2003

desenvolvendo soluções

ATOS AUTOMAÇÃO INDUSTRIAL LTDA.
Rua Arnoldo Felmanas, 201
Vila Friburgo - São Paulo – SP
CEP 04774-010

Departamento Comercial:
Tel: 55 11 5521 5044 - Fax: 55 11 5522 5089
e-mail: comercial@atos.com.br

Fábrica / Assistência Técnica / Engenharia:
Tel: 55 11 5522 1944 - Fax: 55 11 5686 9194
e-mail: atos@atos.com.br

Call Center: 55 11 5522 1944 - Ramal: 111
e-mail: suportec@atos.com.br

ATOS na Internet: www.atos.com.br

Este manual não pode ser reproduzido, total ou parcialmente, sem autorização por escrito da ATOS.

Seu conteúdo tem caráter exclusivamente técnico/informativo e a ATOS se reserva no direito, sem qualquer aviso prévio, de alterar as informações deste documento.

Informações sobre este Manual

• Convenções utilizadas

Listas de itens que não possuam relação direta entre si, iniciam por um marcador "■".

Lista de subitens que não possuam relação direta entre si, iniciam por um marcador "-".

Etapas seqüenciais para a conclusão de um procedimento, iniciam por um número e um ponto.

Títulos de capítulos aparecem grifados com uma barra espessa.

Um capítulo pode ter várias seções. Os títulos dessas seções são precedidos pelo símbolo "•" e grifados com uma linha simples.

Uma seção pode ter várias subseções. Os títulos dessas subseções são grifados com uma linha dupla.

Uma subseção pode ter várias divisões. Os títulos dessas divisões são sublinhados.

Os títulos de figuras e de tabelas são precedidos pelas palavras "Figura" ou "Tabela" mais um número seqüencial.

As referências cruzadas são feitas da seguinte forma: (Veja a página 10), por exemplo.

Palavras em outras línguas são apresentadas entre aspas (" "). Porém algumas palavras são empregadas livremente por causa de sua generalidade e freqüência de uso. Como por exemplo as palavras software e hardware.

Informações que merecem especial atenção por parte do usuário para sua própria segurança ou para o bom funcionamento do equipamento são destacadas com uma moldura de linha dupla.

Os nomes das teclas são escritos entre os caracteres "<" e ">". O caractere "+", entre os nomes das teclas, indica que as mesmas devem ser pressionadas simultaneamente.

• Manuais Relacionados

Manual de Instalação de Controladores Programáveis.

Manual do SUP - Editor de Diagramas de Relés para Conjunto de Instruções DWARE.

Manual do PEP - Programador de EPROM.

Manual do Conjunto de Instruções DWARE.

Manual de Comunicação.

Manual do Controlador Programável MPC4004

Manual do Controlador Programável MPC1600

Manual de Fontes e Módulos de Entradas/Saídas

Manual 162X.Y2 - Terminal IHM LCD GRAPHIC

Manual do ETVG - Editor de Telas Vision Graphic

Manual de Confecção de Gráficos

• Revisões deste Manual

Revisão 1.00 - Abril / 97.

- Revisão inicial do Manual.

Revisão 1.10 - Fevereiro / 98

- Revisão do Texto sobre Supervisório e Redes.
- Revisão dos Exercícios.
- Exercícios de Redes.

Revisão 1.20 - Setembro / 99

- Tópicos novos e exercícios novos

Revisão 1.30 - Março / 03

- Revisão do Lay out

Índice

INFORMAÇÕES SOBRE ESTE MANUAL 3

- **Convenções utilizadas 3**
- **Manuais Relacionados 4**
- **Revisões deste Manual 4**

1. METODOLOGIA DE DESENVOLVIMENTO DE SOFTWARE 7

- **Modularização 7**
- **Estruturação 9**
- **Fluxogramas 9**
  - Processo 9
  - Condição 9
  - Página 9
  - Label 10
- **Diagrama de Tempos e Eventos 12**
- **Tabela de Eventos e Estados 12**
- **Outras considerações 12**

2. ESPECIFICAÇÃO DE CONTROLADORES PROGRAMÁVEIS 13

- **Especificando Entradas e Saídas Digitais 13**
  - Especificando Entradas Digitais 13
  - Especificando Saídas Digitais 13
- **Especificando Entradas e Saídas Analógicas 13**
  - Especificando Leitura de Temperatura 14
  - Especificando Entradas Analógicas 14
  - Especificando Saídas Analógicas 14
- **Especificação Final de Hardware 14**

3. TIPOS DE INTERRUPÇÃO 15

- **Introdução 15**
  - Interrupção 1 15
  - Interrupção 2 15

4. UTILIZAÇÃO DE ENCODER NA CPU 17

- **Modo Normal 17**
- **Modo Ângulo 19**

5. CONTROLE DE TEMPERATURA 21

- **Controle em malha aberta e em malha fechada 21**

- **Controle através do algoritmo PID 22**
  - Descrição do algoritmo PID 22
    - Ação proporcional 22
    - Ação integral 23
    - Ação derivativa 23

- **Ajuste dos Parâmetros do PID2 ..... 24**
  - Ajuste de Kp 24
  - Ajuste de Ki 24
  - Ajuste de Kd 24

- **Análise das Curvas Típicas de um Sistema de Controle de Temperatura 24**

- **Instrução PID 27**

6. COMANDO DE SERVOMOTORES 29

- **Modo Posição 29**

- **Modo Velocidade 31**

7. ACIONAMENTO DE MOTOR DE PASSO 33

- **Visão Geral 33**
  - Estados internos relacionados 33
  - Registros relacionados 33

8. CANAL DE COMUNICAÇÃO SERIAL 37

- **Envio de caracteres através do canal serial (instrução PRINT) 37**
  - Objetivo 37
  - Estados internos relacionados 37

- **Leitura de caracteres através do canal RS232 ..... 37**
  - Objetivo 37
  - Registros e estados internos relacionados 37

9. REDE DE CP'S E SUPERVISÓRIOS 39

- **Rede de CP's 39**
  - Modo Mestre 39

- **Softwares de Supervisão de Processos 40**

10. EXERCÍCIOS 43

1. METODOLOGIA DE DESENVOLVIMENTO DE SOFTWARE

Existem diversas técnicas e métodos que foram criados para auxiliar o desenvolvimento de softwares. Algumas delas podem ser perfeitamente aplicadas à programação em ladder. A seguir tem-se uma abordagem bem superficial sobre alguns conceitos e técnicas que poderão facilitar o desenvolvimento futuro de rotinas em ladder.

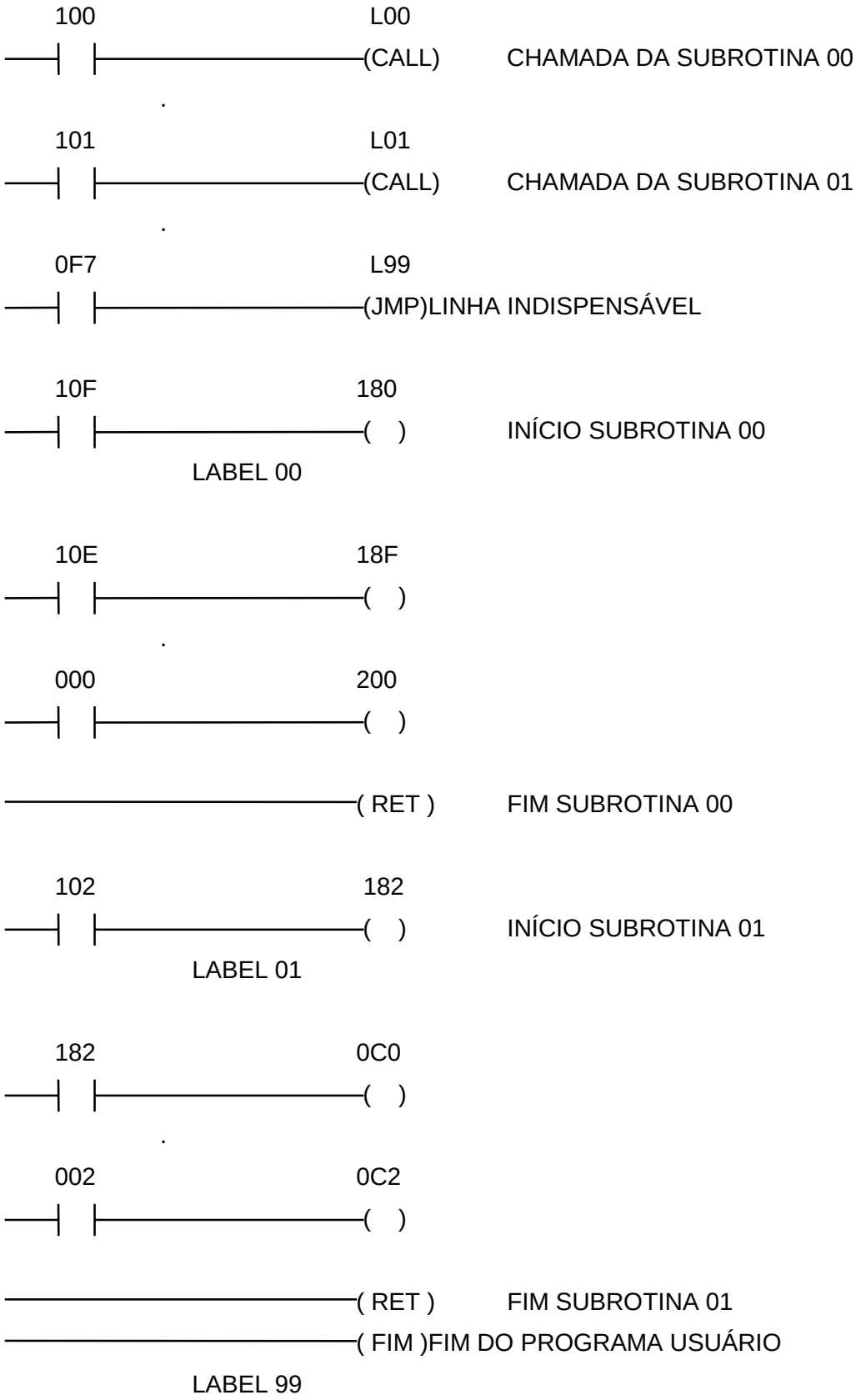
• Modularização

A modularização consiste na divisão do programa em subrotinas, cada uma com uma função específica. Desta forma fica mais fácil para quem analisa o programa entender e alterá-lo quando necessário.

A outra grande vantagem da modularização é poder aproveitar mais facilmente rotinas de um programa para outro. Algumas rotinas como estatística e aquecimento são praticamente idênticas em qualquer programa e podem ser adaptadas facilmente quando usadas como rotinas isoladas.

A estrutura de subrotinas é montada utilizando-se das instruções CALL, RET e JMP.

Exemplo:


100
L00
(CALL)
CHAMADA DA SUBROTINA 00
101
L01
(CALL)
CHAMADA DA SUBROTINA 01
0F7
L99
(JMP)LINHA INDISPENSÁVEL
10F
180
( )
INÍCIO SUBROTINA 00
LABEL 00
10E
18F
( )
000
200
( )
(RET)
FIM SUBROTINA 00
102
182
( )
INÍCIO SUBROTINA 01
LABEL 01
182
0C0
( )
002
0C2
( )
(RET)
FIM SUBROTINA 01
(FIM)FIM DO PROGRAMA USUÁRIO
LABEL 99

• Estruturação

Um programa quando bem montado é simples de ser analisado por qualquer pessoa, um dos principais conceitos que contribuem para isso é a estruturação. Podemos a grosso modo chamar de estruturado um programa que não possui muitas instruções JMP, uma vez que elas causam desvios no programa dificultando a sua análise; observe que a instrução CALL não provoca isso, uma vez que após a instrução RET a execução do programa continua a partir do ponto onde havia sido desviado.

• Fluxogramas

Os Fluxogramas são uma das maneiras de se representar a lógica de um programa ou rotina. Eles consistem em blocos funcionais representando determinadas tarefas que quando ligados juntos formam a lógica do programa. Os principais blocos são:

Processo

Bloco que indica uma ação genérica, como ler uma entrada analógica, realizar uma operação aritmética.

Figura A - Bloco que Representa um Processo

Condição

É um bloco que representa uma condição a ser avaliada, se ela for verdadeira é executada uma lógica se for falsa outra.

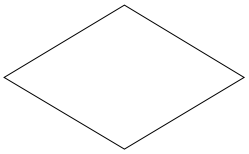


Figura B - Bloco que representa uma condição

Página

Indica que a lógica continua em outra página, usada quando não é possível colocar toda lógica numa única página.

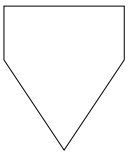


Figura C - Bloco de Fim de Página

Label

É um bloco usado para indicar o início e o fim do programa ou início de uma nova página

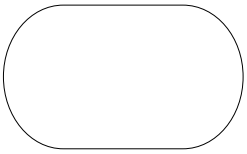


Figura D - Bloco de Label para Início de Programa ou Página

Dessa forma através de uma associação destes blocos básicos temos como montar uma representação da lógica de nossa rotina.

Exemplo: Uma esteira rolante transporta três peças de tamanhos diferentes, o tamanho dessas peças é determinado pelos sensores 1, 2, e 3, de forma que quando a peça 1 é transportada o sensor 1 é atuado e quando a peça 3 é transportada os 3 sensores são acionados. Para cada peça transportada o valor de uma saída analógica que controla a velocidade de um motor é diferente. Existe um último sensor que quando acionado indica que a peca chegou ao fim da esteira e ela deve ser desligada.

O fluxograma para este exemplo ficaria:

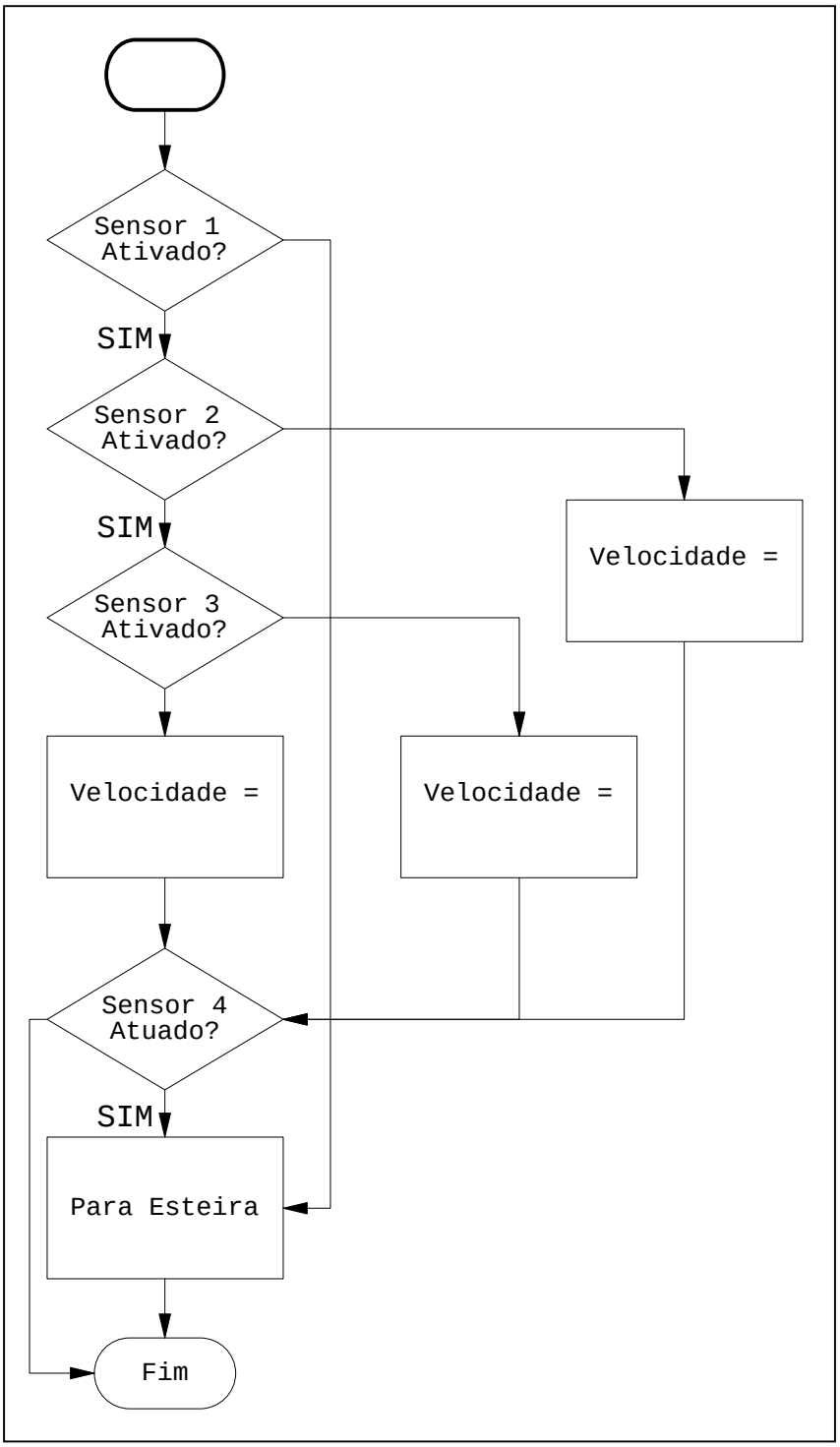


Figura E - Exemplo de Fluxograma

• Diagrama de Tempos e Eventos

O diagrama de tempos e eventos é uma representação, num diagrama temporal das entradas, sinais auxiliares e das respectivas saídas que eles geram. É como se pudéssemos filmar as saídas a cada instante e depois colocar este filme sobre uma folha de papel onde analisaríamos todos os seus valores a cada instante. A seguir temos a representação do diagrama de tempos e eventos para o exemplo anterior.

Para montar o Diagrama de tempos e Eventos criamos linhas que representam cada uma das saídas de nosso processo, e então colocamos sobre esta linha as entradas e suas combinações e indicamos o valor de cada saída. Os estados auxiliares são montados observando-se as entradas e as saídas a cada instante. A seguir temos uma representação para o exemplo anterior de um diagrama de tempos e eventos.

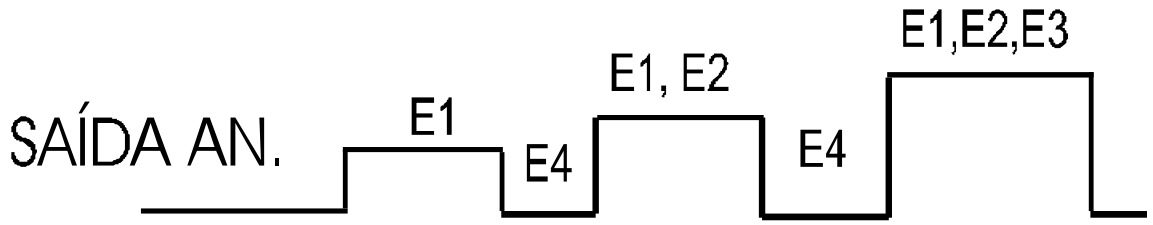


Figura F- Exemplo de Diagrama de Tempos e Eventos

• Tabela de Eventos e Estados

Na tabela de Eventos e Estados, o processo é dividido em etapas, monta-se então uma tabela onde cada etapa é uma linha e cada entrada, saída ou estado auxiliar é uma coluna, assim pode-se determinar para cada combinação de entradas o valor das saídas. Abaixo temos a tabela para o exemplo anterior.

	SENSOR 1	**SENSOR 2**	**SENSOR 3**	**SENSOR 4**	**SAÍDA ANALÓGICA**
LIGA	SIM	NÃO	NÃO	NÃO	V. MÍNIMA
ESTEIRA	SIM	SIM	NÃO	NÃO	V. MÉDIA
	SIM	SIM	SIM	NÃO	V. MÁXIMA
DESLIGA	NÃO	NÃO	NÃO	SIM	V. ZERO
ESTEIRA					

Figura G - Exemplo de Tabela de Eventos e Estados

• Outras considerações

A escolha de como desenvolver uma rotina baseada numa descrição genérica, normalmente leva a mais de uma solução, as ferramentas aqui mostradas servem para auxiliar um pouco e para mostrar o que existe. A prática vai mostrar quando se deve usar cada uma das técnicas apresentadas.

Nosso objetivo era mostrar superficialmente ferramentas, que possam auxiliar o programador, sendo a técnica mais difundida o fluxograma.

2. ESPECIFICAÇÃO DE CONTROLADORES PROGRAMÁVEIS

A especificação correta de controladores programáveis faz com que se obtenha a melhor relação custo benefício do equipamento. Uma vez de posse do descritivo da máquina ou processo devemos responder algumas perguntas para especificar o controlador.

• Especificando Entradas e Saídas Digitais

Especificando Entradas Digitais

O número de entradas digitais deve ser levantado somando-se o número de:

sensores (indutivos, capacitivos, ópticos, etc.)

botões e chaves

outros (térmicos de motores, etc.)

As entradas digitais são especificadas quanto ao nível de tensão:

24 Vdc Tipo P

24 Vdc Tipo N

90 a 240 Vac

Especificando Saídas Digitais

O número de saídas digitais deve ser levantado somando-se o número de:

solenóides

contatores

relés

outros (lâmpadas, sirenes, etc.)

As saídas digitais são especificadas quanto ao nível de tensão:

24 Vdc Tipo P

24 Vdc Tipo N

90 a 240 Vac

relés

• Especificando Entradas e Saídas Analógicas

As entradas e saídas analógicas são usadas para medir e controlar sinais que ao contrário do digital possuem diversos níveis.

Especificando Leitura de Temperatura

Os tipos de termopar são especificados de acordo com a faixa de temperatura do sistema:

Tipo J	0	500 ^{o}C
Tipo K	0	1200 ^{o}C
Tipo S	0	1650 ^{o}C

Com o tipo de termopar e o número de zonas de temperatura podemos escolher o módulo de leitura de temperatura adequado.

Os módulos de leitura de temperatura através de PT100 são especificados de acordo com a faixa de temperatura utilizada no processo e da distância entre o transdutor e o módulo de leitura:

Temperaturas de 0 a + 200 °C

Temperaturas de -50 a +150 °C

Temperaturas de -50 a +50 °C

Conexões a dois fios, até aproximadamente 5 metros

Conexões a três fios, não existe restrição a distância

Especificando Entradas Analógicas

Os sinais de entrada analógica podem ser provenientes de diversas fontes como:

Transdutores de pressão

Transdutores de PH

Réguas Potenciométricas

Estes sinais são encontrados normalmente numa das representações abaixo:

0 a 10Vdc (o sinal pode sofrer influência da resistência ohmica do cabo)

0 a 20mA

4 a 20mA

Especificando Saídas Analógicas

Os sinais de saída analógica podem atuar em diversas fontes como:

Drivers de Potência

Inversores de Freqüência

Amplificadores para Válvulas Proporcionais

Estes sinais são encontrados normalmente numa das representações abaixo:

0 a 10Vdc (o sinal pode sofrer influência da resistência ohmica do cabo)

0 a 20mA

4 a 20mA

• Especificação Final de Hardware

Com essas informações e baseado nos módulos existentes em cada família podemos selecionar o controlador e a configuração que atendem a nossa necessidade com a melhor relação custo/benefício.

3. TIPOS DE INTERRUPÇÃO

• Introdução

A interrupção é uma maneira de se obter uma resposta rápida a uma determinada situação. Ela funciona como uma subrotina, provocando um desvio na execução normal da varredura do programa de usuário. Porém ela não é chamada através de uma instrução CALL como uma subrotina comum, ao invés disso ela é executada toda vez que uma determinada situação ocorre.

Os controladores da ATOS das possuem dois tipos de interrupção:

Interrupção 1 (por hardware ou também chamada por evento)

Interrupção 2 (por tempo)

Interrupção 1

A execução do programa de interrupção 1 ocorre sempre que um determinado evento físico ocorre, como o acionamento de uma entrada digital, o valor de uma entrada analógica atingir um setpoint. Ela tem como objetivo, conseguir do controlador programável o menor tempo de resposta possível, independente do ciclo de varredura desse controlador.

Interrupção 2

O programa de interrupção 2 permite a sincronização de eventos no tempo, uma vez que sempre que ela estiver habilitada sua execução se dá a cada espaço de tempo pré-determinado em milisegundos, independente do tempo de varredura do programa de usuário.

4. UTILIZAÇÃO DE ENCODER NA CPU

O contador rápido no módulo de processamento possibilita o controle de posição através da contagem de pulsos, provenientes de um encoder com saídas **coletor aberto** ou **push-pull**, com uma freqüência máxima de 3 KHz.

Podemos utilizar este recurso de dois modos:

Modo Normal

Modo Ângulo

• Modo Normal

A cada pulso recebido, um registro de contagem é incrementado e uma comparação é executada com um valor de preset pré-determinado pelo usuário, sendo o resultado da comparação colocado em um estado interno que pode ser usado no programa usuário e pode também ser associado a uma saída física configurada pelo programador.

Existem duas formas de se configurar este recurso:

Pelo menu Configuração, submenu Hardware, opção Contador Rápido no Modo Normal, onde serão solicitados três endereços de saídas físicas. (Aplicativo SUP)

Pela edição de pelo menos uma instrução CTCPU no software usuário:

Esta instrução faz a comparação do efetivo do contador rápido presente na CPU com o conteúdo de um registro (OP1). Pode-se escolher que tipo de comparação será executada (efetivo >, < ou = ao conteúdo de OP1) e quando o resultado da comparação for verdadeiro será acionado um único estado interno (OP2) em cada bloco.

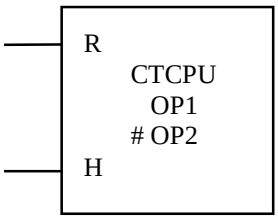
No mesmo programa usuário só poderão haver oito instruções CTCPU, das quais, somente uma habilitada por vez. O estado interno 3FF quando acionado indica valor "negativo" no efetivo do contador.

A instrução possui duas entradas:

RESET - quando ativada permite o reset do efetivo do contador rápido presente na CPU.

HABILITA - quando acionada a instrução é executada.

Símbolo em diagrama de relés:



Observações: " # " poderá ser >, < ou =;

A contagem irá de 0 a 99999999;

Não existe sinal " Z ".

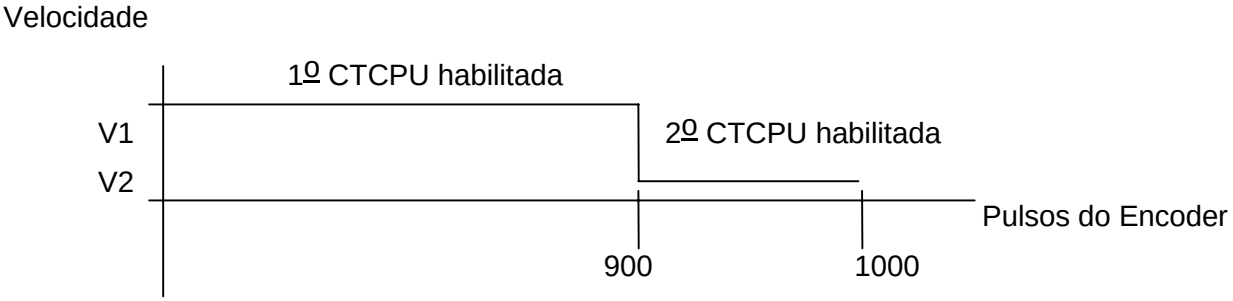
Exemplo de aplicação:

Esta aplicação demonstra a utilização da instrução CTCPU em uma máquina que faz o avanço de um filme plástico, onde esse avanço é feito em duas etapas:

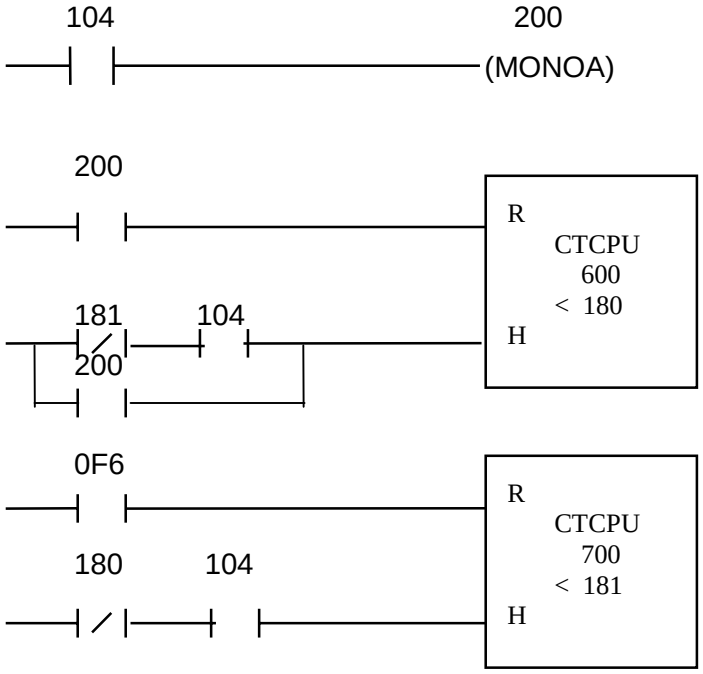
- etapa 1 com avanço rápido (velocidade alta).
- etapa 2 com avanço lento (velocidade baixa).

Supondo que cada avanço seja de 1000 mm (pulsos do encoder), a troca de velocidade alta para velocidade baixa é feita em 900 mm.

Observe o gráfico:



Deste modo teremos o trecho de programa abaixo:



Quando a entrada 104 for acionada, o estado interno 200 também será acionado durante uma varredura (através da instrução MONOA), deste modo o efetivo do contador será resetado. Na próxima varredura a saída 180 será acionada e assim permanecerá enquanto o efetivo for menor que o conteúdo do registro 600 (neste caso 600=0000 e 602=0900). Com o desacionamento da saída 180 ocorrerá a habilitação da segunda instrução CTCPU e da mesma forma a saída 181 ficará acionada até que o efetivo do contador fique com um valor igual ou superior ao conteúdo do registro 700 (neste caso 700=0000 e 702=1000).

• Modo Ângulo

Este modo exige a utilização de um encoder angular que fornecerá os pulsos que serão comparados com 16 regiões definidas através de presets iniciais e finais, caso o número de pulsos esteja compreendido entre estes presets, ocorrerá o acionamento de um estado interno relacionado a esta região.

O funcionamento do modo ângulo pode ser comparado ao funcionamento de um came, os pulsos enviados pelo encoder seriam a referência de velocidade de giro deste came, cada uma das 16 regiões citadas, poderia ser associada a um ressalto em cada um dos 16 discos do referido came e os estados internos relacionados às regiões seriam as chaves fim de curso que existiriam no came.

Podemos configurar este recurso pelo menu Configuração, submenu Hardware, opção Contador Rápido no Modo ângulo. (Aplicativo SUP)

Mapeamento de memória:

Modo ângulo (1)

04DB 04DA	VALOR DA MARCA ZERO PARA SENTIDO DECRESCENTE
04D9 04D8	VALOR DA MARCA ZERO PARA SENTIDO CRESCENTE
04D7 04D6	EFETIVO
04D5 04D4	RESERVADO
04D3 04D2	RESERVADO
04D1 04D0	VALOR EM RPM **(2)**

(1) No modo ângulo o valor da marca zero para sentido decrescente é igual ao número de pulsos por volta menos um.

(2) O cálculo do valor em RPM é feito só no modo ângulo e considerando-se encoder de 360 pulsos/volta, independentemente do encoder realmente usado.

003F 0030	16 EI DE ÂNGULOS (Modo Ângulo)

Estrutura de dados para ângulos iniciais e finais:

051E	ÂNGULO FINAL 08
051C	ÂNGULO INICIAL 08
051A	ÂNGULO FINAL 07
0518	ÂNGULO INICIAL 07
0516	ÂNGULO FINAL 06
0514	ÂNGULO INICIAL 06
0512	ÂNGULO FINAL 05
0510	ÂNGULO INICIAL 05
050E	ÂNGULO FINAL 04
050C	ÂNGULO INICIAL 04
050A	ÂNGULO FINAL 03
0508	ÂNGULO INICIAL 03
0506	ÂNGULO FINAL 02
0504	ÂNGULO INICIAL 02
0502	ÂNGULO FINAL 01
0500	ÂNGULO INICIAL 01

053E	ÂNGULO FINAL 16
053C	ÂNGULO INICIAL 16
053A	ÂNGULO FINAL 15
0538	ÂNGULO INICIAL 15
0536	ÂNGULO FINAL 14
0534	ÂNGULO INICIAL 14
0532	ÂNGULO FINAL 13
0530	ÂNGULO INICIAL 13
052E	ÂNGULO FINAL 12
052C	ÂNGULO INICIAL 12
052A	ÂNGULO FINAL 11
0528	ÂNGULO INICIAL 11
0526	ÂNGULO FINAL 10
0524	ÂNGULO INICIAL 10
0522	ÂNGULO FINAL 09
0520	ÂNGULO INICIAL 09

Exemplo:

Seja o primeiro ângulo inicial de 0º e primeiro ângulo final de 150º:

0500h	0501h
00	**00**

0502h	0503h
01	**50**

5. CONTROLE DE TEMPERATURA

O controle de temperatura nos controladores programáveis, obedece um algoritmo chamado PID (Proporcional, Integral e Derivativo). Para entender melhor este algoritmo alguns conceitos devem ser abordados

• Controle em malha aberta e em malha fechada

Num sistema em malha aberta temos um sinal de entrada x(t) também chamado de setpoint (valor que desejamos obter numa grandeza física), e um sinal de saída y(t) também chamado de variável de processo que vai atuar dependendo exclusivamente desta entrada. A figura abaixo representa um controle em malha aberta.

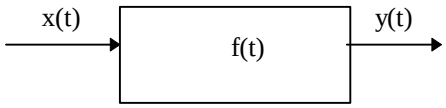


Figura H- Sistema de controle em malha aberta

Num sistema em malha fechada, existe, além do sinal de entrada x(t) (setpoint), um retorno do sistema que realimenta o circuito, fazendo com que se possa medir o erro (diferença entre a variável de processo e o setpoint). A saída passa então a ser função deste erro e não mais da entrada simplesmente, fazendo com que o controle do sistema seja mais eficaz.

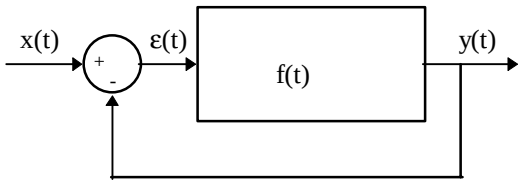


Figura I- Sistema de controle em malha fechada

Onde:

x(t) = Valor desejado na saída (setpoint ou preset SP)

y(t) = Valor que controla a saída (variável de processo PV)

ε(t) = Erro do sistema (x(t) - y(t))

f(t) = função que aplicada ao erro gera y(t)

●Controle através do algoritmo PID

Definições:

- SETPOINT OU PRESET - temperatura programada no controlador , é a temperatura que se quer atingir.

- EFETIVO - temperatura lida pelo cartão do CP, temperatura real.

- OVER-SHOOT - é a maior temperatura registrada, sendo atingida no início do aquecimento do processo.

- DESVIO OU ERRO DO SISTEMA - é a diferença entre o setpoint e o valor efetivo .

- BANDA - região onde ocorrerá o controle de temperatura; Abaixo da banda, as resistências estão ligadas, e acima da banda desligadas.

- TEMPO **-** valor em segundos, para cálculo do período da saída PWM, desta forma se tivermos um tempo de 4 segundos, com uma saída igual a 50%, teremos a saída digital 2 segundos ligada (Ton) e 2 segundos desligada (Toff) .

Descrição do algoritmo PID

O algoritmo PID pode ser escrito de maneira simplificada, conforme a equação:

S = P + I + D onde:

S - saída para controle do processo , podendo ser analógica ou do tipo PWM (Pulse Width Modulation)

O algoritmo PID , é a soma dos três elementos , que combinam suas ações , para executar o controle da variável do processo (temperatura).

O usuário poderá definir a contribuição de cada parâmetro programando ganhos para cada um dos termos, os quais são descritos abaixo:

Kp - ganho proporcional (0% a 100 %)

Ki - ganho integral (4 a 250 repetições / minuto)

Kd - ganho derivativo (0 a 25,5 minutos)

Ação proporcional

O controle proporcional mantém uma relação linear entre o valor da variável de processo e a posição do elemento final de controle.

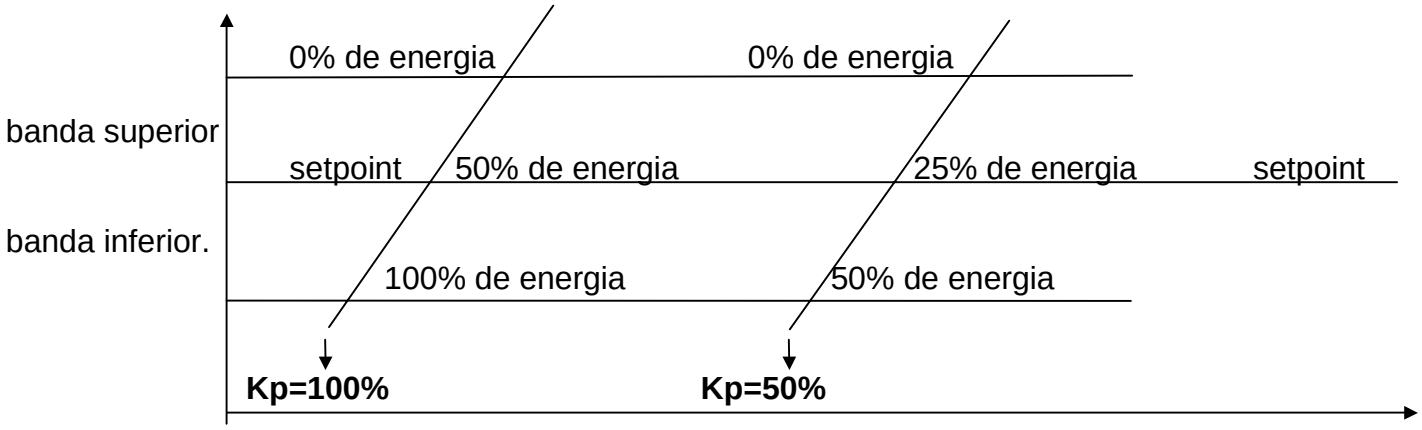
A magnitude da correção é proporcional à amplitude do desvio, ou seja, a saída do controlador é proporcional ao erro.

Quanto maior for o desvio, maior será a correção do termo proporcional

A unidade empregada será de porcentagem, podendo variar de 00 a 100% o termo proporcional.

O gráfico abaixo dá uma melhor noção da influência do termo proporcional.

supondo: S = P (controle somente com termo proporcional)



Ação integral

A finalidade da ação integral é eliminar o desvio permanente deixado pela Ação Proporcional, provocando a contínua correção do sinal de saída até que o erro seja eliminado.

A correção é proporcional à integral do erro.

Enquanto existir desvio a saída do controlador irá aumentar ou diminuir, só cessando a variação da saída quando o desvio desaparecer.

O termo integral pode ser expresso como a quantidade de repetições (somas do erro) ocorridas por unidade de tempo. A unidade empregada é REPETIÇÕES / MINUTO, podendo executar desde 4 a 250 repetições por minuto.

A contribuição do termo integral poderá ser positiva ou negativa, desta forma a soma de P+I poderá alcançar o valor máximo para a saída (100%) , ou mínimo (0%) , tendo como referência o setpoint.

Ação derivativa

O termo derivativo introduz uma ação corretiva proporcional a velocidade de variação do desvio.

Combinada com a Ação Proporcional faz com que, quando a variável de processo se afasta do setpoint, a saída varie mais do que variaria somente com a ação P ou P+I.

Por outro lado quando a variável está retornando ao valor original, o Modo Derivativo exerce uma ação contrária, reduzindo as eventuais oscilações. Pode se dizer que a finalidade da Ação Derivativa é diminuir o tempo de correção do desvio, antecipando a ação corretiva.

A Ação Derivativa é também conhecida por ação antecipatória, e o tempo de antecipação é chamado “tempo derivativo”, e é expresso em minutos.

•Ajuste dos Parâmetros do PID2

No algoritmo do PID2, o termo proporcional (Kp), é o responsável pela energia média entregue à carga. Quando ajustado em 100%, no setpoint ele fornece 50% de energia para a carga, no limite inferior da banda ele fornece 100% e no limite superior ele fornece 0%. Se Kp=50% temos 25% no setpoint, 50% no limite inferior e 0% no superior.

O termo integrativo (Ki) pode variar de 4 a 250 repetições por minuto. Ele pode contribuir com uma faixa de ± 50% da energia na carga, que em conjunto com o termo proporcional permite obter uma variação de 0 a 100% de energia entregue à carga.

O valor de Ki determina quantas vezes por minuto é calculado o erro do sistema (setpoint - valor real), este erro vai alterar o valor da parcela integral de energia, aumentando ou diminuindo o valor da energia na carga de forma a diminuir o erro.

O termo derivativo tem a função antecipatória, ou seja, ele tenta "prever" como um determinado erro vai se propagar ao longo do tempo e corrige a energia na carga de maneira a minimizá-lo. Para isso ele faz a diferença entre as duas últimas medidas de temperatura. Caso a temperatura esteja aumentando ele vai diminuir a energia na carga, caso a temperatura esteja diminuindo ele vai aumentar a energia. A contribuição máxima que ele pode dar é de ± 25% de energia.

Na maioria dos casos, os valores padrão de Kp, Ki, e Kd serão capazes de estabilizar o sistema na temperatura correta, contudo um ajuste sobre os parâmetros pode diminuir o tempo de estabilização e o over-shoot inicial.

Ajuste de Kp

Kp deve ser aumentado caso a temperatura se estabilize abaixo do setpoint.

Kp elevado provoca maior over-shoot inicial e temperatura estável acima do setpoint.

Ajuste de Ki

Ki muito alto pode fazer com que o sistema não estabilize, oscilando em torno do setpoint.

Ki alto provoca correção do erro mais rápida, porém provoca maior over-shoot

Ajuste de Kd

Kd elevado provoca menor over-shoot, diminuindo o tempo de estabilização, porém o sistema pode oscilar em torno do setpoint.

• Análise das Curvas Típicas de um Sistema de Controle de Temperatura

Um sistema de controle de temperatura consiste em fornecer energia através de resistências de forma a compensar as perdas de calor do sistema. Esta energia é controlada pela modulação no tempo do acionamento das resistências.

O sistema de controle de temperatura consiste em determinar o erro, a velocidade de variação do erro , o erro médio acumulado e reagir aumentando ou reduzindo o tempo de acionamento das resistências de forma a eliminar estes erros.

Um sistema ideal de controle de temperatura, elevaria a temperatura na máxima velocidade até o setpoint. Retiraria e forneceria energia ao sistema para mantê-lo no setpoint.

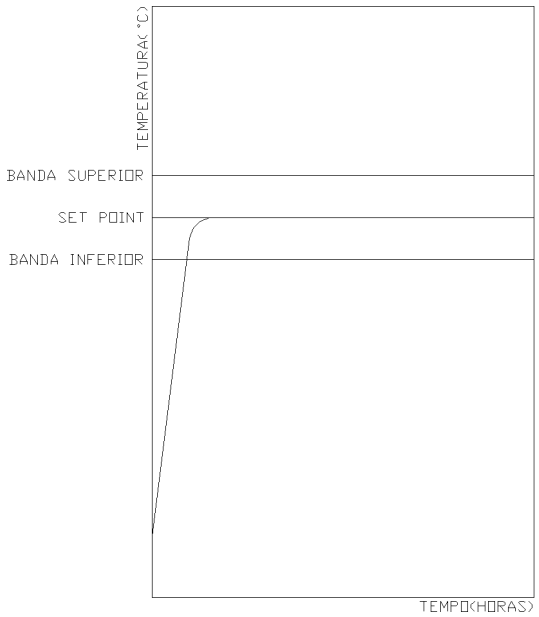


Figura J- Sistema ideal de controle de temperatura

Normalmente, os sistemas não possuem uma forma eficiente de retirar calor. Portanto devemos alcançar o setpoint com uma velocidade inferior, levando um tempo maior.

.

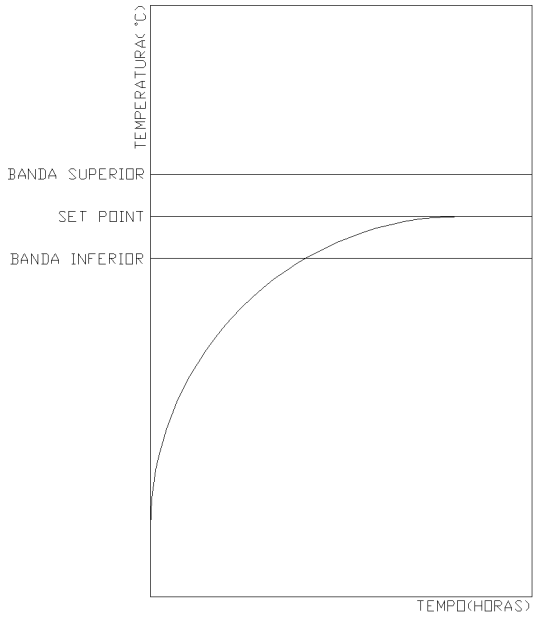


Figura K- Sistema ideal com baixa velocidade

Outra solução é fixar uma região de atuação em torno do setpoint, de forma que abaixo desta região a energia fornecida seria máxima (resistências em ON), e acima não forneceria (resistências em OFF). Esta região de atuação, convencionou-se chamar de banda (B). Este tipo de atuação irá provocar um efeito chamado de over-shoot, mas por outro lado deixará o sistema mais sensível em torno do setpoint.

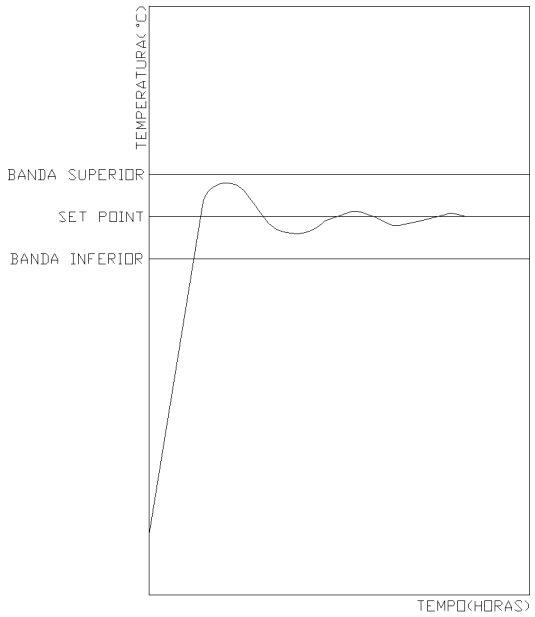


Figura L- Típico Over-shoot

ATENÇÃO: Aumentando-se a banda reduz-se o over-shoot e a sensibilidade e vice-versa.

Após o over-shoot em um sistema em malha aberta, a temperatura irá se estabilizar em torno do ponto onde a energia fornecida pelas resistências seja igual a energia perdida pelo sistema. Esta energia pode ser determinada por Kp.

Energia média = Kp. Emax .

onde Emax = Máxima energia das resistências.

Como o sistema possui uma inércia natural entre a variação da energia (resposta do sistema) e a temperatura lida pelo termopar, a simples correção do erro instantâneo, em casos de variações bruscas, faz com que o sistema sinta e corrija o erro tarde demais. Com isto a temperatura oscila muito em torno do setpoint.

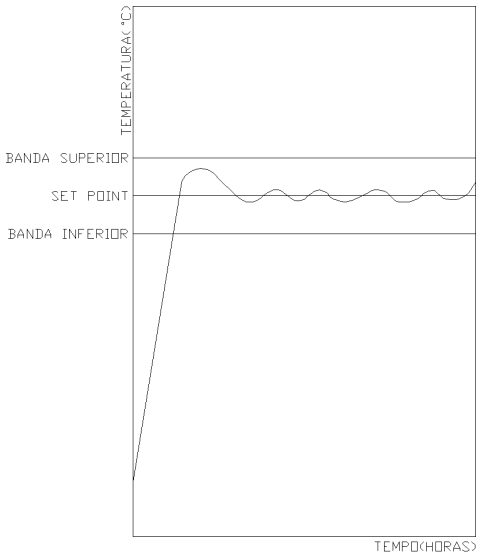


Figura M- Sistema apenas com controle proporcional

Neste caso a ação derivativa sentiria uma brusca variação da temperatura e a corrigiria antes que o erro se torne grande.

Em sistema PD onde é fornecida energia muito diferente da energia média consumida para manter a temperatura em um determinado setpoint, pode ocorrer que a temperatura fique estável, porém com uma certa diferença em relação ao setpoint, pois a energia que a ação proporcional fornece para corrigir o erro, apenas o atenua e como a temperatura permanece estável a ação derivativa não atua .

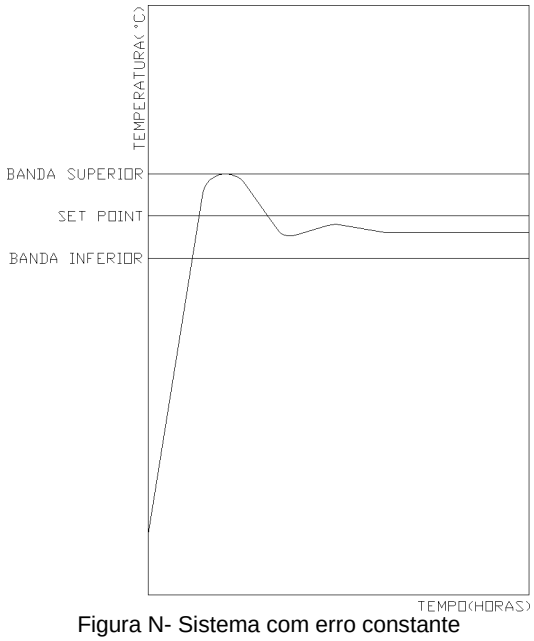


Figura N- Sistema com erro constante

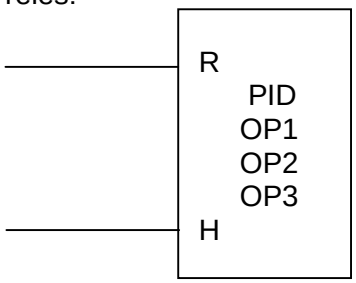
No sistema PID isto seria corrigido, pois em casos de erros acumulados a ação integral faria uma correção crescente até eliminar este erro.

• Instrução PID

Esta instrução contém três operandos, uma entrada Habilita e uma entrada Reset para carregar o valor inicial da somatória do erro.

Quando a entrada Habilita (H) é acionada e a entrada Reset (R) estiver acionada, o valor inicial da somatória do erro é transferido para a região de parâmetros e efetuado o cálculo do PID (deve ser utilizado um estado interno gerado por uma saída MONOA na entrada Reset (R), caso contrário o valor inicial da somatória do erro será transferido para a região de parâmetros toda varredura). Se a entrada Habilita (H) é acionada e a entrada Reset (R) estiver desacionada, é efetuado apenas o cálculo do PID.

Símbolo em diagrama de relés:



onde:

OP1 - Variável de entrada (efetivo)	(0 - 9999)
OP2 – Setpoint	(0 - 9999)
OP2 + 02 – Banda	(0 - 255)
OP2 + 04 - Kp (ganho proporcional)	(0 - 100%)
OP2 + 06 - Ki (ganho integral)	(4 - 250 rep/min)
OP2 + 08 - Kd (ganho derivativo)	(0 - 25,5 min)
OP2 + 0A - RESERVADO	
OP2 + 0C - Valor mínimo da saída	(0 - 1000)
OP2 + 0E - Valor máximo da saída	(0 - 1000)
OP2 + 10 – Tempo	(0 - 25,0 seg)
OP2 + 12 - Estado Interno de Aquecimento	
OP2 + 14 - Valor inicial somatória do erro	(0 - 9999)
OP2 + 16 - RESERVADO	
OP2 + 18 - RESERVADO	
OP2 + 1A - RESERVADO	
OP2 + 1C - RESERVADO	
OP2 + 1E - RESERVADO	
OP3 - Variável de saída	(0 - 1000)

OBS: O valor inicial da somatória do erro pode ser positivo ou negativo. Os valores positivos vão de 0 a 4999, e os negativos de 5000 a 9999.

Se não for efetuado o carregamento do valor inicial da somatória do erro, teremos um valor indefinido no mesmo.

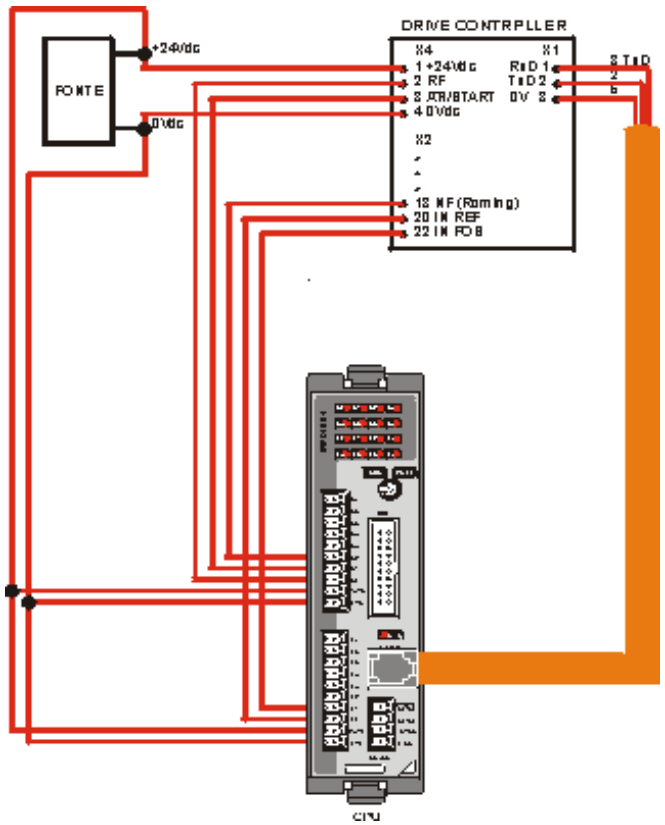
6. COMANDO DE SERVOMOTORES

• Modo Posição

O Modo Posição ou Modo Posicionamento é indicado para casos onde não seja necessário mudar parâmetros (posição, velocidade, aceleração, etc.) durante a execução do movimento.

Exemplo disto é a aplicação de um servo-motor para acionar um dosador rosca numa máquina de embalagem.

Este modo consiste em enviar os parâmetros posição, velocidade, aceleração e outros através de um dos canais seriais RS232 ou RS485 para o Drive de Controle, os parâmetros tendo sido enviados pode-se então partir o servo-motor por meio de sinais digitais. Deste modo se o Drive de Controle estiver devidamente parametrizado o eixo do servo-motor será posicionado com precisão.



Detalhes para implementação do software usuário:

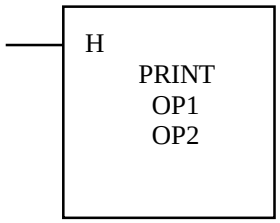
- Protocolo de comunicação do equipamento que será controlado.
- Formato dos comandos e dos valores dos parâmetros (Hexadecimal/ASCII).
- Necessidade ou não do cálculo de checksum.
- Caso o canal seja o RS485 implementar retardo entre os envios.

- Número de eixos que serão comandados.

Instrução responsável pelo envio de dados para o servo-drive:

Os dados que serão enviados são os bytes contidos entre os operandos OP1 e OP2. A instrução tem uma entrada Habilita, necessitando também para ser ativada que um estado interno (0FB em ON escolhe impressora) esteja ligado.

Símbolo em diagrama de relés:



É necessário que o formato dos dados seja ASCII e que OP1 seja menor que OP2.

Exemplo de programação:

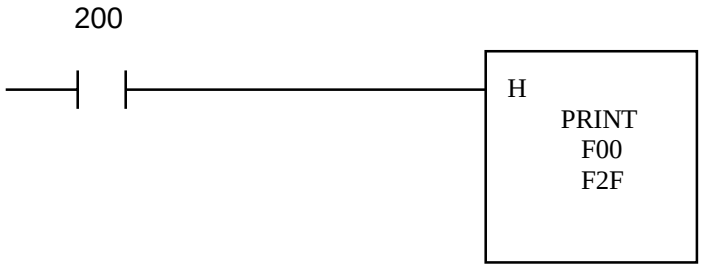
Para a impressão de:

ATOS
AUTOMAÇÃO INDUSTRIAL

Neste formato (centralizado) em uma impressora serial de 32 colunas deve-se carregar o seguinte bloco de dados :

F00=2020 F02=2020 F04=2020 F06=2020 F08=2020 F0A=2020 F0C=2020 F0E=4154 F10=4F53
F12=0A0D F14=0A0D F16=2020 F18=2020 F1A=2020 F1C=4155 F1E=544F F20=4D41 F22=4341
F24=4F20 F26=494E F28=4455 F2A=5354 F2C=5249 F2E=414C

O registro F00 foi escolhido como início do bloco de dados.

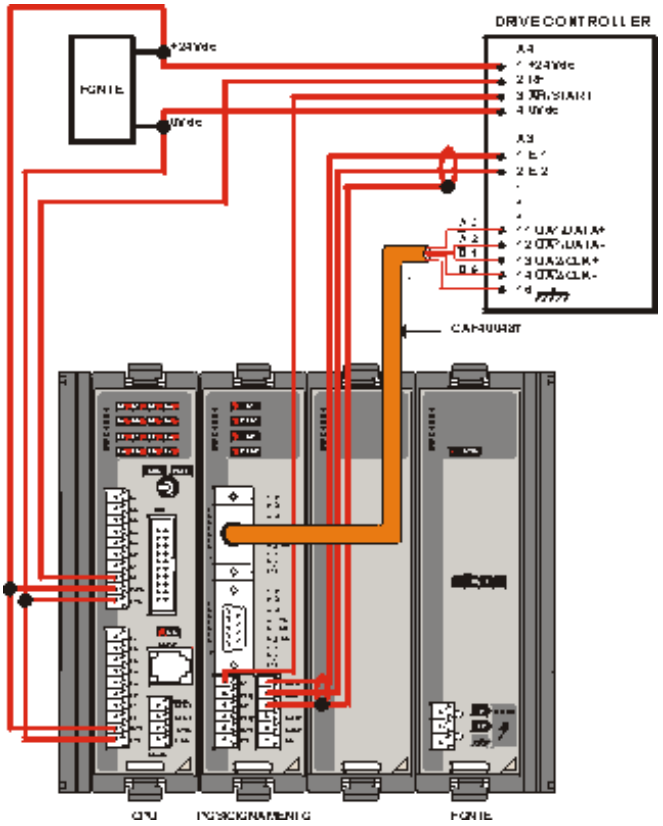


Deste modo se o estado interno especial 0FB estiver acionado (indicando impressora), quando o estado interno 200 for acionado através de MONOA ou MONOD, o bloco de dados acima será transferido para o canal serial do CP utilizado.

• Modo Velocidade

O Modo Velocidade é indicado para casos onde seja necessário mudar constantemente a velocidade do eixo do servo-motor durante a execução do movimento.

Exemplo disto é a aplicação de um servo-motor para puxar o filme plástico numa máquina de “corte e solda” (máquina que produz sacolas para super mercados).Neste caso a estratégia é, através de uma saída analógica, controlar a referência de velocidade do servo-motor com relação a posição do seu eixo, que será monitorada pelo CP através do simulador de encoder do servo-drive. Deste modo assim que é dado início de puxada do filme devemos ter uma referência de velocidade diferente de zero para que o servo-motor vença a inércia da carga e comesse a se mover, consequentemente o CP receberá pulsos com a informação de posição do eixo (comprimento do filme já tracionado) e a partir daí a velocidade se comportará de acordo com os cálculos feitos pelo CP para rampas de aceleração e desaceleração com base na velocidade máxima, para se conseguir a melhor performance possível da máquina.



A instrução SCL2G é a responsável pelos cálculos que resultarão no controle da velocidade durante a puxada do filme.

7. ACIONAMENTO DE MOTOR DE PASSO

• Visão Geral

A família MPC4004 permite o acionamento de 1 motor de passo de 4 fases X 2 A , podendo ser ligado diretamente nas saídas do controlador .

As saídas utilizadas são : 180 a 183 , sendo as demais, 184 a 187 , de uso geral .

Para ativar o modo motor de passo, é necessário habilitar o "modo motor de passo" no menu configuração submenu hardware, nesta condição os seguintes registros e estados internos são válidos:

Estados internos relacionados

200 - habilita torque - quando ativado ira energizar o motor com o último passo ativo.

201 - bloqueio - quando ativado inibirá a progressão de contagem , parando o motor instantaneamente deixando torque no eixo do motor.

202 - escolha do modo de funcionamento :

Ativado - modo contínuo

Desativado - modo posição

Modo contínuo - nesta condição após a habilitação do motor , o mesmo começará a girar indefinidamente.

Modo posição - nesta condição , o motor se deslocará uma quantidade programada de pulsos , parando com torque no final da contagem.

203 - sentido ativado horário , desativado anti-horário

Obs: a direção de rotação está relacionado a seqüência de pulsos que o motor irá receber , desta forma para mudar a direção de rotação basta inverter a seqüência de acionamento das fases o motor .

204 - posição alcançada . Este estado interno será ligado toda vez que o motor estando no modo posição e após ser habilitado, atingir a posição definida nos endereços 4D8/4DB.

205 - escolha do tipo de passo : desligado - passo inteiro

ligado - meio passo

Obs: a escolha de meio passo permite dobrar a resolução do motor .

206- Reset do efetivo - Ao ser ativado colocará zeros nos endereços 4D4/4D5 e 4D6/4D7

Registros relacionados

4D4 e 4D6 - efetivo de contagem dos passos (8 dígitos) (modo posição)

4D8 e 4DA - preset do número de passos (8 dígitos) (modo posição)

4D0 - valor de velocidade min. 5.0 RPM máx 180.0 RPM

OBS: A velocidade do motor em RPM calculada considerando um motor de 360 passos por volta .

A velocidade máxima efetivamente alcançada depende exclusivamente do tipo de motor que se está utilizando , bem como do torque necessário ao processo (Quanto mais veloz menor será o torque do motor).

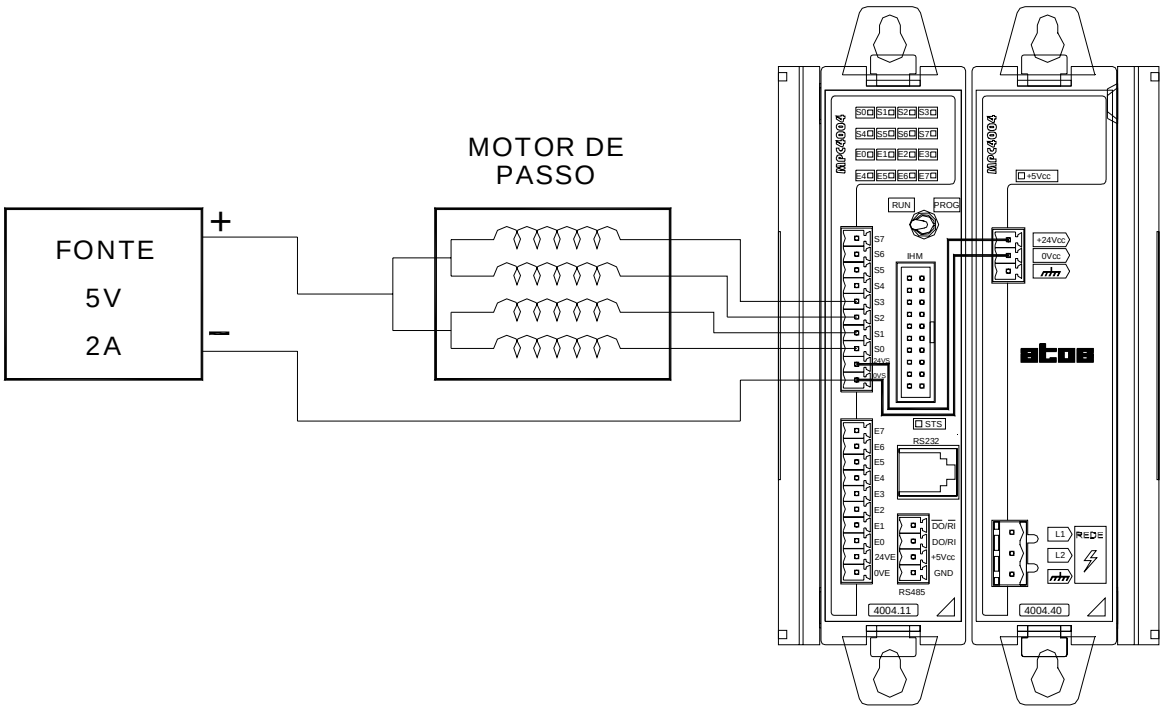
tabelas de acionamento:

Passo inteiro				
	S180	S181	S182	S183
1	ON	OFF	ON	OFF
2	ON	OFF	OFF	ON
3	OFF	ON	OFF	ON
4	OFF	ON	ON	OFF
1	ON	OFF	ON	OFF

Meio passo				
	S180	S181	S182	S183
1	ON	OFF	ON	OFF
2	ON	OFF	OFF	OFF
3	ON	0FF	OFF	ON
4	OFF	OFF	OFF	ON
5	OFF	ON	OFF	ON
6	OFF	ON	OFF	OFF
7	OFF	ON	ON	OFF
8	OFF	OFF	ON	OFF
1	ON	OFF	ON	OFF

Interligação física com o controlador programável

Esquema de ligação :



As saídas do controlador programável podem ser divididas em dois módulos :

circuito de controle e circuito de potência .

O circuito de controle necessita de uma tensão de 24V /10mA para funcionar .

O circuito de potência pode ser considerado um circuito com coletor aberto , permitindo a conexão de tensões que podem variar de 3 a 30V com correntes de até 2 A .

Desta forma , uma vez polarizado o circuito de controle em 24V , a ligação do motor de passo nas saídas poderá ser feito com tensões mais baixas por ex. 5V , sem a necessidade de limitadores de tensão .

Importante: A tensão aplicada às fases do motor de passo é sempre constante , independente da velocidade do motor , portanto não é recomendado trabalhar em baixas velocidades por muito tempo em função do aquecimento do motor. Ë recomendável limitar o valor mínimo da velocidade em pelo menos 20RPM .

Caso as saídas digitais do CP tenham que ser do tipo “P” o motor de passo terá que ser alimentado com a mesma tensão da fonte externa 24 Vdc.

8. CANAL DE COMUNICAÇÃO SERIAL

• Envio de caracteres através do canal serial (instrução PRINT)

Objetivo

Permitir que o usuário possa enviar caracteres para um dispositivo externo como uma impressora serial, servo motores , modem etc

Estados internos relacionados

0FB - Habilita modo Print

0BD - Determina para qual canal serial serão enviados os caracteres

desligado - canal RS232, ligado - canal RS485

0FC - estado interno que indica canal serial ocupado ou seja durante a transmissão dos dados ele ficará ligado. Este estado auxilia o usuário a sincronizar o envio de várias mensagens.

Funcionamento: A o habilitar o modo print (EI 0FB ligado), e selecionado o canal a ser enviado , o usuário deverá ativar a instrução "PRINT" através de um "MONOA para enviar os dados através do canal serial.

Importante: O estado 0FB deve ficar ativo durante todo o tempo de transmissão dos dados .

Ao ativar o estado 0FB , o controlador não mais poderá receber programação através do SUP, pois seu canal serial fica reservado para o envio de dados .

A taxa de transmissão para o modo print é definida pelo usuário no menu de configuração de hardware .

• Leitura de caracteres através do canal RS232

Objetivo

Permitir que o usuário possa ler caracteres de um dispositivo externo como leitor de código de barras, servo-motores ,retorno de conexão com modem, etc.

Registros e estados internos relacionados

0AB - Habilita leitura de caracteres do canal RS232

0FB - Habilita modo Print

0E00 - 0EFE - Buffer de recepção dos caracteres (255)

0FE4/0FE5 - Registro contador de caracteres recebidos

Funcionamento: Estando em modo Print (EI 0FB ligado) , e com o estado interno especial "0AB" também ligado, os dados recebidos em RX do canal de comunicação RS232 são armazenados a partir do endereço 0E00 até um limite de 255 caracteres.

Quando o estado "0AB" estiver desligado , os caracteres recebidos em RX do canal de comunicação RS232 são ignorados .

A quantidade de bytes recebidos é atualizada no registro 0fe4/0fe5 .

A transição de off para on do estado interno 0AB,provoca a limpeza do buffer (colocação do valor "FF" entre 0E00 e 0EFF) e zera o registro contador de caracteres recebidos .

funcionamento do estado interno 0AB :

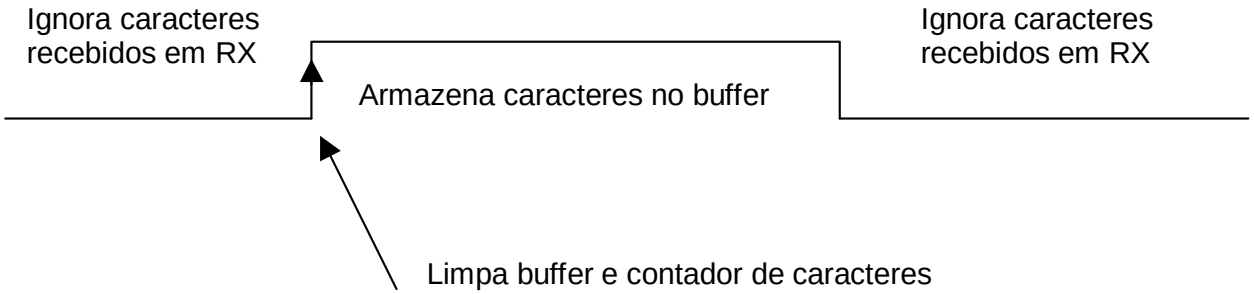

Ignora caracteres recebidos em RX
Armazena caracteres no buffer
Ignora caracteres recebidos em RX
Limpa buffer e contador de caracteres

Fig. - Funcionamento do Estado Interno 0AB

9. REDE DE CP's E SUPERVISÓRIOS

• Rede de CP's

Uma rede entre CP's visa o intercâmbio de informações, possibilitando, por exemplo, que um sistema supervisório tenha acesso a praticamente todas as informações presentes nos equipamentos do "Chão de Fábrica", ou então que se consiga distribuir o hardware de controle, facilitando assim sua instalação e diminuindo bastante o volume da cablagem.

Modo Mestre

Objetivo : Capacitar a família MPC4004 com o recurso de "mestre de rede" no canal RS485, possibilitando a troca de informações entre controladores através da comunicação background.
A programação background é útil por exemplo no transporte de alarmes das estações onde o programador terá, além das informações do processo controlado pelo mestre, as informações das estações supervisionadas .

Estados internos relacionados:

3D0 - Estado interno de habilitação do modo mestre , ao ser ativado o canal serial RS485 iniciará a varrer a tabela com as regiões a serem atualizadas nas estações .
Ao ser desligado, o canal RS485 volta a ser um canal escravo.

Importante : O canal RS485 ao ser definido como mestre não mais responderá a aplicativos como SUP, PEP ou sistemas supervisórios, pois estará havendo colisão no canal de comunicação e função de haver 02 dispositivos mestre na rede .

3D1 a 3EF - indicam respectivamente estados internos de falha de comunicação com as estações de 01 a 31 .
Quando houver mais de 05 tentativas consecutivas sem sucesso com uma determinada estação, será ligado automaticamente o estado interno de falha , sendo desligado automaticamente quando houver o restabelecimento da comunicação.

Regiões de comunicação:

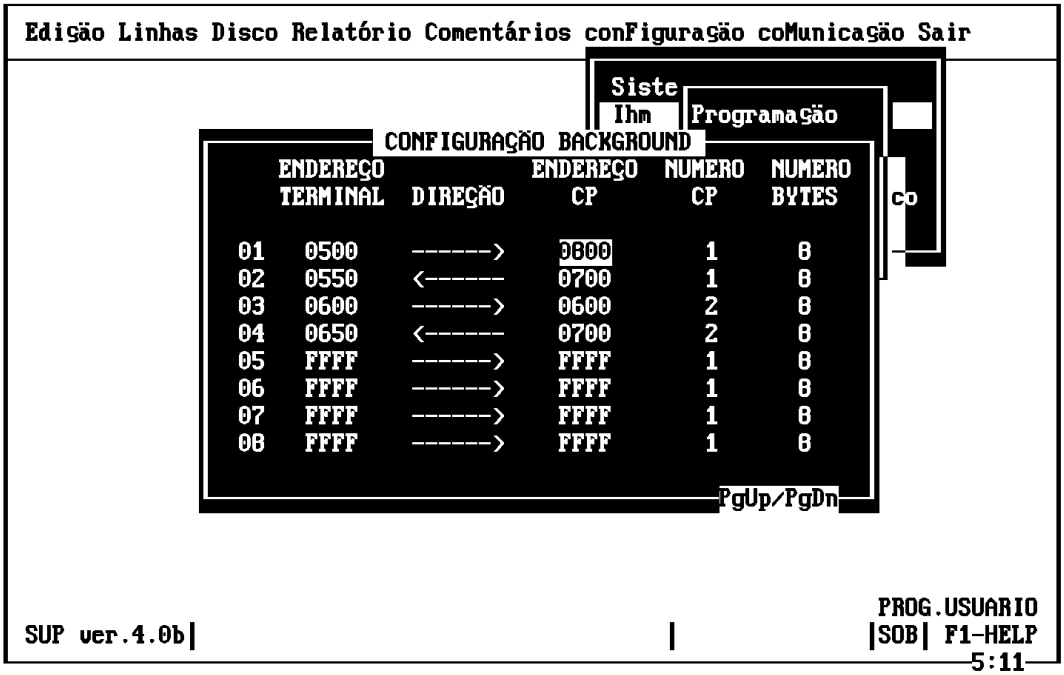
Ë possível definir até 40 regiões de comunicação de 8 bytes cada .
Cada região receberá ainda o atributo de enviar para a estação ou receber da estação:

- "Terminal → CP "- envia os bytes do terminal para o CP
- "Terminal ← CP " - envia os bytes do CP para o terminal

Também é possível definir o endereço do mestre e o endereço do CP onde ocorrerá o envio ou recebimento das informações .

A programação background é definida no SUP no menu Configuração submenu Background.

Exemplo da tela de configuração da comunicação background:



• Softwares de Supervisão de Processos

Os softwares de supervisão de processos, também chamados supervisórios são programas que rodam num PC e se comunicam com o controlador programável por uma porta serial, trocando informações e permitindo uma visualização gráfica do processo. Os supervisórios também permitem que se criem relatórios impressos e gráficos que mostram o comportamento do sistema. Cada ponto de conexão entre o supervisório e o controlador é chamado de "tag", ou seja, um supervisório com 50 "tags", permite que se tenha 50 variáveis do controlador sendo monitoradas.

Os supervisórios permitem que se configurem as telas, permitindo uma visualização gráfica do processo, eles funcionam normalmente em 3 modos:

- Configurador: modo em que é possível a definição das telas, endereços a serem monitorados, gráficos, relatórios. Alguns fabricantes exigem a utilização de hardkey (dispositivo de proteção) para funcionar.
- Runtime: modo de execução da aplicação desenvolvida no modo configurador, não permite a edição dos parâmetros e necessita de um Hardkey (dispositivo de proteção) para funcionar.
- Demo: modo limitado que permite normalmente alguns minutos de comunicação com o CP, na maioria das vezes limitando também o número de tags que se pode utilizar.

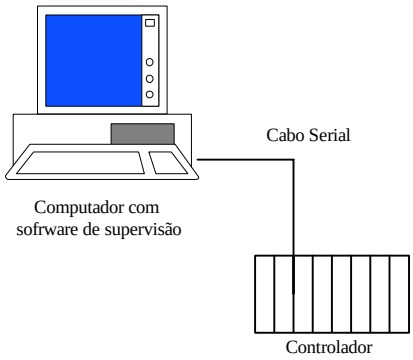


Figura -Supervisório ligado ao Controlador

A seguir está relacionado os softwares de supervisão que possuem driver de comunicação com os controladores da ATOS.

SOFTWARE DE SUPERVISÃO	TELEFONE
CITECT - Citect Brasil	(035) 622 1311
ELIPSE - Elipse Software Ltda.	(011) 814 3746
FIX DMACS - SPA Sistemas Planejamento e Análise	(011) 284 6844
GENESIS – COPRODIN & INOVARE Sistemas de Computação	(011) 542 6088
INTOUCH - Scan Automação Ltda.	(011) 814 0863
IPC-XPERT - Rifran Eletrônica Ltda.	(011) 5564 7522
UNISOFT - Unisoft Ltda.	(011) 532 2870
WIZCON - Soft Brasil.	(011) 885 6381

10. EXERCÍCIOS

1 Uma máquina de corte de filme plástico possui um inversor de freqüência que controla a velocidade e a parada do motor que puxa o filme, o comprimento do filme que será cortado é monitorado através de um encoder. A puxada do plástico deve ser feita em duas etapas, uma em velocidade alta e outra em velocidade baixa. Desenvolva o software usuário para controlar esta máquina.

2 Faça o controle de uma zona de temperatura, prevendo os alarmes de termopar aberto, invertido, temperatura abaixo da mínima, acima da máxima. Implemente uma rotina de pré-aquecimento com programação para todos os dias da semana e horário de início e fim do pré-aquecimento, utilize preset de 60 C para o pré-aquecimento e 70 C para o controle normal, deve haver uma seletora para ativar o pré-aquecimento e outra para o aquecimento através da IHM.

3 Existe um alimentador de chapa para uma prensa que utiliza um servo-motor para avançar a chapa para debaixo da prensa, devemos controlar o servo-motor através dos parâmetros posição e velocidade, onde a posição corresponde ao comprimento da chapa e a velocidade com que o avanço será feito. Utilize o envio destes parâmetros através do canal serial do CP.

4 Num sistema de usinagem de ranhuras numa peça cilíndrica, temos um motor de passo posicionando a ferramenta que faz as ranhuras na peça, um sensor indica que a peça esta na posição para início do processo que leva um determinado tempo, em seguida o motor de passo desloca a ferramenta a partir do número de passos presetado pelo operador, o número de vezes que estas etapas ocorrem também é determinada pelo operador através da interface. Configure o recurso de receitas para este processo, contendo preset de passos e ranhuras para cada receita.

5 Monte um relatório de produção contendo identificação numérica do operador, quantidade de peças produzidas, este relatório será impresso por uma impressora serial conectada diretamente ao CP, a impressão será disparada a partir de uma das teclas da interface e neste instante deverão ser amostradas a data e o horário que farão parte do relatório.

6 Desenvolver um sistema de controle de acesso para um estacionamento, onde teremos um leitor de código de barras conectado ao CP que deverá acionar ou não uma cancela. Faça isto através da leitura de dados pelo canal serial do CP.

7 Um motor têm sua velocidade controlada por uma saída analógica, tendo como velocidade 0 rpm correspondente a 0 V e 3600 rpm correspondentes a 10 V, no eixo deste motor existe um sensor de velocidade que faz um retorno de velocidade para o controlador, correspondendo a 0 à 3600rpm a faixa de 0 à 10V. A aceleração do motor deve ser feita gerando-se uma rampa, com um número de degraus especificado pelo operador e a desaceleração do mesmo modo. Existe ainda um botão no frontal que quando pressionado uma vez liga o motor, pressionando-o novamente desliga-se o motor. Especifique o hardware necessário para esta aplicação.

8 Desenvolver um controle de aquecimento de um tanque contendo um líquido que deve ser mantido aquecido a 120 ^{o}C. A temperatura mínima e máxima do sistema são 80 e 150 ^{o}C respectivamente. Caso essas temperaturas sejam atingidas deve-se emitir alarmes na IHM, deve-se gerar alarmes também em caso de termopar aberto ou invertido. Deve-se ter uma seletora de liga aquecimento, via frontal, sem a qual não se emitem os alarmes ou se aquece o sistema. Deve-se ajustar os parâmetros PID do sistema. Especifique o hardware necessário para esta aplicação.

9 Num sistema de posicionamento, existe um encoder bidirecional acoplado ao eixo gerando pulsos ao controlador. Sabendo que cada pulso corresponde a 1 mm e o tamanho do eixo de posicionamento é de 10m, fazer uma rotina de posicionamento que a um comando do operador realize um deslocamento da posição atual para a programada. Existem ainda dois botões que ao serem acionados levam o sistema para o início ou o fim do eixo. Especifique o controlador que atende a esta aplicação Sugestão: usar os botões dos frontais.

10 Um sistema de envasamento de garrafões de água tem a capacidade de encher 3 garrafões ao mesmo tempo. Para isso existe uma esteira que transporta os garrafões até que o acionamento do sensor 1 indicar que eles estão na posição para enchimento. Nesse instante a esteira é parada e abre-se a válvula de enchimento dos garrafões, durante um tempo determinado. Os garrafões podem ser de 5, 10 ou 20 litros, com tempo de enchimento de 10, 20 ou 40 segundos, respectivamente. Criar um contador de 8 dígitos do número de ciclos produzidos, criar também um predeterminador de produção fazendo que quando ativado seja envasado o número escolhido de garrafões e depois parado o processo. Criar também uma tela no frontal com a hora e a data do sistema (usar a instrução BMOVX). O sistema deve funcionar em manual, automático e semi-automático (apenas um ciclo executado). Especifique o controlador necessário. Obs.: Usar os botões do frontal e criar arquivos de moldes para os tipos de garrafão. Colocar a lógica do totalizador e predeterminador numa subrotina. Desenvolva também de acordo com suas necessidades as telas para a interface gráfica.

11 Deseja-se elaborar um controle de iluminação e incêndio de um edifício comercial de 10 andares. Cada andar possui 2 salas e 2 corredores. Deve-se também automatizar o bombeamento de água dos reservatórios inferiores para os superiores. Cada sala terá um sensor de luminosidade analógico e 3 sensores digitais de incêndio. Cada corredor terá um sensor de cada.

O controle de iluminação é feito desligando-se parte das luminárias quando o nível natural for maior que 300 lux no corredor e 600 lux nas salas.

Após a última parada deve-se desligar os elevadores e abrir suas portas.

O controle do bombeamento de água do reservatório inferior para o superior deve ser noturno, exceto quando o nível do reservatório superior estiver abaixo do mínimo de operação.

Desenvolver o controle para um andar e para o reservatório de água.

Existe um software de supervisão que monitora o processo.

Obs.: carregar o programa em dois CP's e montar uma rede.